Hindawi
Security and Communication Networks
Volume 2023, Article ID 1164728, 9 pages
https://doi.org/10.1155/2023/1164728

Research Article

A Novel High-Efficiency Password Authentication and Key Agreement Protocol for Mobile Client-Server

Xiang Xu, Yu Wang, Xue-Dian Zhang, and Min-Shan Jiang

Shanghai Key Laboratory of Contemporary Optics System, College of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China

Correspondence should be addressed to Min-Shan Jiang; jiangmsc@gmail.com

Received 30 August 2022; Revised 12 March 2023; Accepted 16 March 2023; Published 17 April 2023

Academic Editor: Chien Ming Chen

With the development of wireless technology, people increasingly rely on mobile devices. Since most mobile devices transmit sensitive information via insecure public channels, it is important to design multiauthentication key agreement protocols for security protection. Traditional scholars tend to use traditional public-key cryptosystems (PKCs) in their protocols to improve security. High-cost operations (e.g., elliptic curve point multiplication and bilinear pairing) were widely used in their scheme but were not suitable for mobile devices because of limited computing resources. In this study, we designed a novel high-efficiency multiauthentication and key agreement protocol and demonstrate its security in the random oracle model. Compared with other protocols, our proposed scheme only uses string concatenation operations, one-way hash functions, and XOR operations. In addition, our protocol requires much fewer computing resources to achieve the same level of security.

1. Introduction

In recent decades, wireless network mobile devices have been applied to various scenarios (e.g., wireless payment systems, instant communication, and remote authentication) with the development of technology. Compared with traditional devices, mobile devices are more flexible. People can use mobile devices to pay, receive messages, and perform other tasks, regardless of when and where they are. This technological revolution improves people's quality of life, and mobile devices are expected to encompass a wide variety of uses.

However, the mobile device transmits some sensitive information via an insecure public channel, often at great risk. The Kaspersky Lab reported that cybercriminals easily commit crimes with small investments, due to the high mobile bank usage (Brazilian Federation of Banks statistics show Brazil's mobile bank usage reached more than 11.2 billion transactions with 33 million active accounts in 2015) and the low cost of short message service (SMS) messages [1]. Figure 1 briefly describes a communication model of the mobile client-server environment. Because of the opening of the environment, attackers can replay, modify, or intercept messages and try to pretend as legitimate users/servers to complete authentication or access the user's sensitive information. To protect user privacy, it is crucial for us to design a secure multi-authentication and key agreement protocol for mobile terminals.

There are plenty of password authentication and key agreement protocols proposed to protect users' privacy in open network environments. Generally, these traditional protocols are based on hard mathematic problems, such as ECC [2, 3] (relies on elliptic curve discrete logarithm problem (ECDLP)), RSA [4] (relies on the integer factorization problem (IFP)), and Elgamal [5] (relies on the discrete logarithm problem (DLP)).

The high-cost modular exponentiation operation is widely used in these public key cryptography (PKC) [6, 7]. Therefore, it is impractical for mobile devices to use traditional protocols due to insufficient central processing unit (CPU) power and random access memory (RAM). To address these problems, Boneh and Franklin [8] proposed an ID-based protocol using elliptic curves. However, the

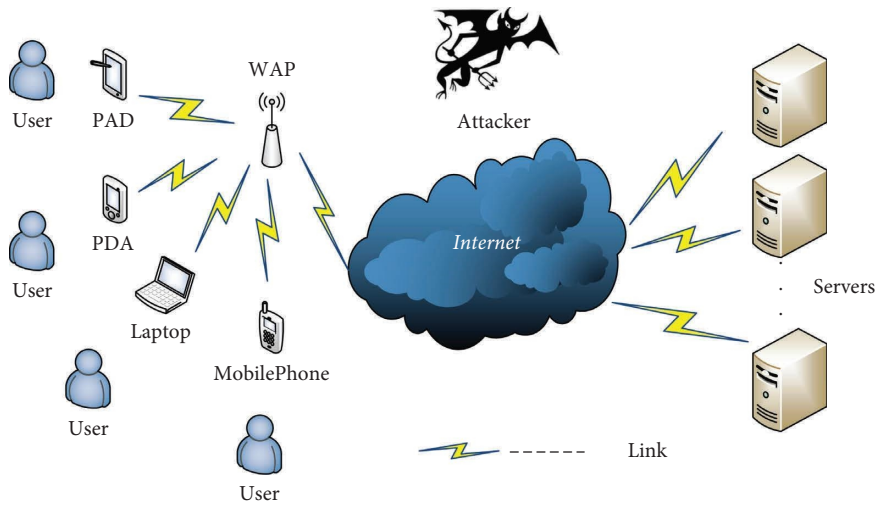

Figure 1: Communication model.

protocol using ECC was still not efficient enough for mobile devices, because the computationally expensive and time-consuming elliptic curve point multiplication operations were often a burden to the mobile device. In addition, some scholars chose bilinear pairings in their schemes to ensure security and provide better performance to some extent. The aforementioned schemes were unsuitable for mobile devices because bilinear pairings are a computationally expensive and time-consuming operation. Therefore, designing a secure and efficient authenticated key agreement protocol (AKAP) for mobile terminals to protect users' privacy is critical.

In this article, we described designing a simplified AKAP for the mobile terminal without any kinds of complex calculations (e.g., elliptic curve point multiplication operations, modular multiplication operations, bilinear pairing operations). The protocol we proposed only uses some simple operations, such as string concatenation operations, one-way hash functions, and exclusive OR (XOR) operations. Compared with previous works, the highlights of our proposed protocol are summarized as follows:

(i) Our protocol does not employ any complex operations that require a large amount of computational resources. Hence, the proposed protocol is able to work on different types of mobile terminals.

(ii) The security of our protocol is demonstrated in the random oracle model.

(iii) Our protocol is more effective and secure. As a result, most mobile devices can use this protocol.

The remainder of our article is organized as follows. In the section "Related Works," the related works are presented. In the section "Our scheme," the specifics of the proposed scheme are illustrated. Then, the proposed scheme is proven to be secure under the random oracle model in the section "Security Proof." In the section "Comparison and performance analysis," the evaluation result of the proposed scheme is discussed. Finally, the conclusion is given.

2. Related Works

In 1981, the first single server environment authentication protocol was presented by Lamport [9]. The traditional authentication protocol for a single-server architecture is incapable of directly applying to multiple-server architectures. In 2001, Li et al. [10] first designed multiple-server architecture authentication based on a neural network, but did not perform well for the sake of network complexity. In 2004, Juang [11] proposed a new protocol adopting symmetric cryptography that was unable to defend against insider attacks. Afterwards, some new protocols [12, 13] (adopting symmetric cryptography) were designed to increase security. For the abovementioned protocol, the client ID in the message was shown in plaintext. Consequently, their scheme could not protect the anonymity of users' identities, especially in wireless networks. In 2009, Liao and Wang [14] used symmetric cryptography to design a dynamic ID-based scheme for privacy protection. Hsiang and Shih [15] found the problem that Liao and Wang [14] protocol could not defeat masquerade attacks, insider attacks, server spoofing, and registration center spoofing attacks. In addition, their protocol was unable to support mutual authentication. Hsiang and Shih's [15] presented an enhanced scheme to overcome these weaknesses. Later, Lee et al. [16] detected that Hsiang and Shih's protocol [15] was unable to withstand a server spoofing attack. Since then, numerous dynamic ID-based protocols using symmetric cryptography have been proposed for different application environments. It could not, however, reflect user obscurity and unlinkability. Debiao et al. [17] developed an elliptic curve-based validation technique in 2012. Wang and Ma [18] revealed that it did not provide mutual authentication and was vulnerable to refection attacks. Farash and Attari [19] offered an ECC-based mutual authentication and key exchange mechanism for a mobile client-server scenario. The user is authenticated by using his or her user identification and private key, while the server uses its private key. This protocol did not ensure user anonymity.

Jegadeesan et al. [20] offered a mutual authentication mechanism between a mobile user and a service provider that is anonymous. In the registration step, the suggested protocol places a high value on a trusted authority. This is a point of failure in the system. In a multiserver mobile cloud computing (MCC) scenario, Irshad et al. [21] described an authentication approach based on pairing-based cryptography. The usage of a registration authority constitutes a system failure point. Olufemi Olakanmi and Oke [22] presented a mutual authentication mechanism that protected privacy while addressing MCC security problems. The proposed approach combined the voice signature of the user with cryptographic procedures. The usage of a trusted authority constitutes a system failure point. Tsai and Lo [23] offered an anonymous authentication technique for distributed MCC services based on pairing-based cryptography. According to the researchers, their system offered mutual authentication, key exchange, and user untraceability, but their model has a flaw, which is that the fingerprint is misused since the collection of the biometric parameter does not always produce the same value. Furthermore, several researchers [24–26] discovered that this protocol is vulnerable to server spoofing attacks.

In 2012, Debiao et al. [17] suggested an ID-based AKAP using ECC. This protocol was adopted by the random oracle model for mobile client-server(C/S) environments. To achieve efficiency, they selected hash functions instead of ineffective map-to-point functions. However, Hafizul and Biswas [27] found that Debiao et al.'s [17] user in their protocol was also anonymous, and the protocol could not prevent impersonation attacks, insider attacks, or ephemeral information attacks. Sun et al. [28] suggested an AKAP protocol based on ECC for mobile C/S environments in 2013. Their scheme defeated a privileged insider attack using a private key that was commensurate with the server. He and Wang [29] designed an improved protocol based on ECC to address security. Odelu et al. [30] demonstrated that an adversary could easily learn the user's identity in He and Wang's protocol. To address these weaknesses, Odelu et al. [30] suggested a biometric-based multiserver authentication protocol using smart cards. Mo et al. [31] and Tseng et al. [32] proposed a remote ID-based AKAP using ECC for mobile devices.

Recently, Azrour et al. [33] proposed a new Internet of Things (IoT) device authentication protocol. They demonstrated both formally and informally that their protocol was effective and resilient to various attacks. Wazid et al. [34] designed a new lightweight authentication mechanism in the cloud-based IoT environment to prevent information leakage during communication. In 2021, Chaudhry et al. [35] stated that Wazid's protocol was unable to offer mutual authentication between the system elements when there were many registered users. They then proposed an enhanced system and established both its formal and informal security.

3. Our Scheme

To achieve higher efficiency, our proposed protocol replaces the complex operations with some simple operators. Because the proposed protocol only uses lightweight operations, the proposed protocol can work on different types of mobile terminals with high efficiency. Although we do not use complex operations, our proposed protocol is still capable of reaching excellent security levels.

3.1. Notation Phase. User U and server S chose their master key and system parameters at this phase by carrying out the subsequent actions. The server S first chose a secret random number s as the master key. Afterwards, the server S chose a one-way hash function $H(\cdot)$ [36], which maps an arbitrary length string to a l-bit string. Finally, the server S stored the master key s into the database and published the one-way hash function $H(\cdot)$. Table 1 summarizes the notations commonly used in this article.

3.2. Registration Phase. If a person wants to be a legal user and use his or her mobile terminal to accomplish uniform identity authentication, he or she must take the following steps. This phase's details are given below, and the flow chart is presented in Figure 2.

Step $R1$: User U chooses a random number r_i to compute $T_m = H(\mathrm{PW}_m \| r_i)$ and $\mathrm{PID}_m = H(\mathrm{ID}_m \| r_i)$. Then, it sends the message $\{\mathrm{PID}_m, T_m\}$ as a secure channel registration request to server S.

Step $R2$: After receiving the message $\{\mathrm{PID}_m, T_m\}$ from user U, the server S computes $R_m = H(\mathrm{PID}_m \oplus T_m \oplus s)$, $h_m = R_m \oplus h(T_m \oplus w_s)$, and $\mathrm{TID}_m = H(\mathrm{PID}_m \oplus w_s)$ (where w_s represents a random number generated by the server S). Then, server S stores $\{\mathrm{PID}_m, \mathrm{TID}_m, T_m\}$ in the database, and uses a secure channel to deliver message $\{\mathrm{PID}_m, h_m, w_s\}$ to user U.

Step $R3$: After getting the message $\{\mathrm{PID}_m, h_m, w_s\}$, user U computes $R_i = H(ID_m \oplus PW_m) \oplus r_i$. Then, U stores $\{R_i, \mathrm{TID}_m, h_m, w_s\}$ into read-only mobile device.

3.3. Login and Authentication Phase. When user U wants to access server S, user U must perform the steps outlined below. The details are described in Figure 3.

Step $L1$: When the mobile device needs to be authenticated, user U must input his or her ID_m and password PW_m which corresponds to the identity.

After completing the login phase, user U must take the following steps to complete the authentication phase.

Step $A1$: User U selects a secure random number r_0. Then, user U computes $r_i' = R_i \oplus H(ID_m \oplus \mathrm{PW}_m)$, $T_m' = H(\mathrm{PW}_m \| r_i)$, $\mathrm{PID}_m' = H(\mathrm{ID}_m \| r_i)$, $R_m' = h_m \oplus H$

Table 1: Notations, functions, and system parameters.

Notations	Our scheme
S	Server
U	User
s	Master key of the server
ID_m	User U's identity
PW_m	User U's password
$H(\cdot)$	One-way hash function
$\oplus$	Bit exclusive-or operation
$\|$	String concatenation operation

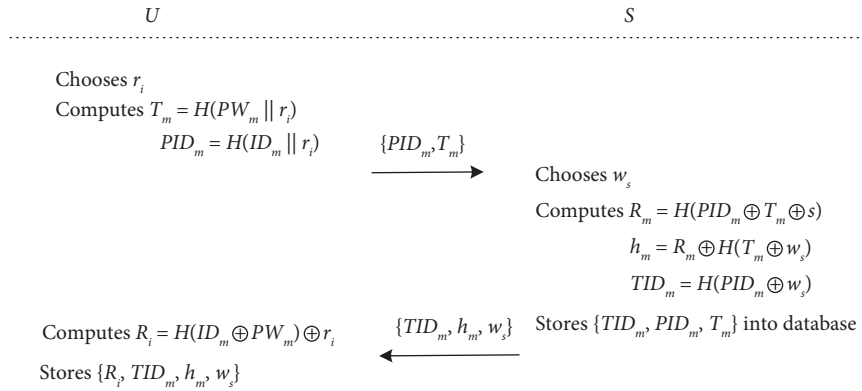

Figure 2: Registration phase.

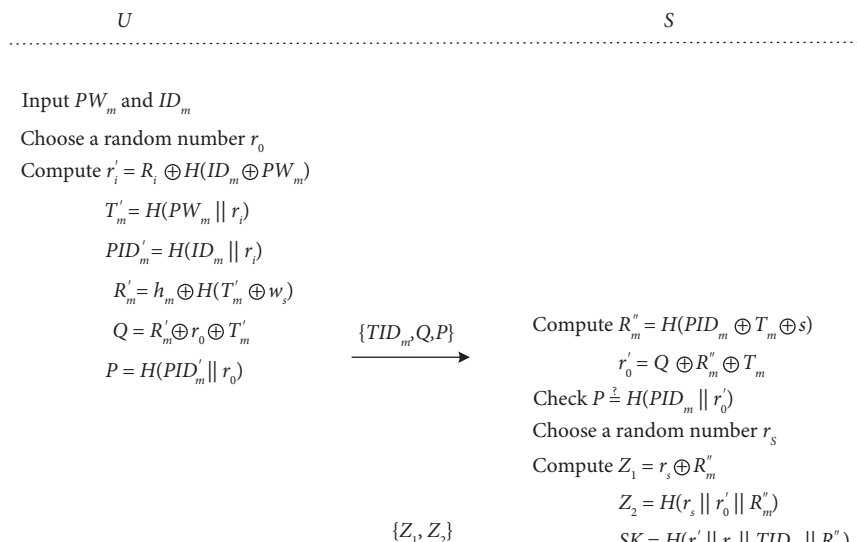

Figure 3: Login phase and mutual authentication phase.

$(T_m' \oplus w_s)$, $Q = R_m' \oplus r_0 \oplus T_m'$, and $P = H(\mathrm{PID}_m' \| r_0)$. Finally, user U sends message $\{\mathrm{TID}_m, Q, P\}$ to server S.

Step $A2$: The server S performs the following steps to identify the user U after receiving the authentication request from user U. First, user U computes $R_m'' = H(\mathrm{PID}_m \oplus T_m \oplus s)$ and $r_0' = Q \oplus R_m'' \oplus T_m$. Then, user U checks whether P is equal to $H(\mathrm{PID}_m r_0')$. If the condition is satisfied, S validates the identity of user U; if not, S denies U's request. Next, S chooses a secure random number r_s. Then, S computes $Z_1 = r_s \oplus R_m''$, $Z_2 = (r_s \| r_0' \| R_m'')$, and $SK = H(r_0' r_s \| \mathrm{TID}_m \| R_m'')$ (where SK is the session key between S and U).

Step $A3$: After receiving the message from S, user U carries out the steps below.

First, the user U computes $r_s' = R_m' \oplus Z1$. Then, it checks whether Z_2 is equal to $H(r_s' \| r_0 \| R_m')$. If the condition holds, S verifies U's validity; otherwise, S denies U's request. Finally, U computes $\mathrm{SK}' = H(r_0 r_s' \| \mathrm{TID}_m \| R_m')$ (SK' is the session key between S and U).

4. Security Proof

In this phase, we demonstrated that the suggested protocol P can resist multiple attacks under the real-or-random model [37]. There are two kinds of participants in our scheme. One is user U, and the other is server S. These definitions are described below:

Adversary: In this security model, the adversary A can control all the communications in our protocol fully and it runs in polynomial time [38]. The following are the specific abilities:

SEND(S, M): Attacker A receives a message M generated by server S as a response after delivering a message to the server S. This query simulates active attacks, such as modification attacks, impersonation attacks, and replay attacks.

EXECUTE(U, S): The message sent by user U to server S is obtained by attacker A. The eavesdropping attack is modeled in this query.

TEST(u/S): This query simulates the semantic security of SK by flipping an unbiased coin c. The instance user u returns a binary of the same size as session key SK if the hidden bit $c = 0$ or the session key SK if $c = 1$. If attacker A asks many Test(u/S) queries, the output should be static.

CorruptMD(u): The attacker A obtains a message stored in $U's$ mobile device when one makes a query. This query simulates a mobile device lost/stolen attack, in which the information contained in mobile device is known via the power analysis attack [39].

Semanticsecurity: The adversary A may engage with the instances to assist her or him in identifying the value of bit b if the above queries are provided. If she or he properly guesses, the system fails to offer semantic security. Let SUCC represent the event in which A succeeds. In breaking the semantic security of the scheme, A has an advantage $\mathrm{Adv}_{P1}^{\mathrm{Ake}}(A) = |2 \Pr[\mathrm{SUCC}] - 1|$. The scheme is safe under the real-or-random model if $\mathrm{Adv}_{P1}^{\mathrm{Ake}}(A)$ is minimal.

Theorem 1. *First, assume that D is a uniformly distributed password dictionary and that A is the adversary running in polynomial time t against our protocol P. Then,*

$$\mathrm{Adv}_{P1}^{\mathrm{Ake}}(A) \leq \frac{q_h^2}{|\mathrm{Hash}|} + \frac{2q_{\mathrm{send}}}{|D|}, \tag{1}$$

where q_h, q_{send}, $|\mathrm{Hash}|$, and $|D|$ indicate the number of hash queries, the quantity of SEND(S, M) *queries, the hash function's range space, and the size of the dictionary D, respectively.*

Proof. A sequence of games Gi (where $i = 0, 1, 2, 3$) are defined in this proof. Then, let SUCC_i be an event wherein attacker A can guess hidden bit c successfully in game Gi.

Game G0: This game model is attacked by the adversary A in the random oracle model, and the hidden bit c is chosen randomly at the beginning of this game. From the above definitions, we have the following equation:

$$\mathrm{Adv}_{P1}^{\mathrm{Ake}}(A) = 2 \Pr[\mathrm{SUCC}_0] - 1. \tag{2}$$

Game $G1$: This game queries oracle EXECUTE(U, S) to simulate the attacker's eavesdropping attack. Finally, A queries the TEST(u/S) oracle and decides whether the value of the hidden bit c in the TEST(u/S) oracle is a random number or the right session key (SK). The session key SK is calculated by r_0', TID_m, r_s, and R_m. Usually, attacker A tries to obtain this message from the public channel. Obviously, the attacker A cannot guess the secret random number r_0' and r_s. Meanwhile, we know that

$$\begin{aligned} R_m &= H(\mathrm{PID}_m \oplus T_m \oplus s) \\ &= H\left(H\left(\mathrm{ID}_m \| r_i\right) \oplus \left(H\left(PW_m \| r_i\right) \oplus s\right)\right). \end{aligned} \tag{3}$$

Therefore, without access to the server's database or the mobile device, attacker A is unable to compute the session key SK. The users' identity and passcode, and the server's master key are still unknown. Finally, we can conclude that attacker A gains both the mobile device and the server's database, and the chance of winning for attacker A is not increased by eavesdropping. Therefore, we have the following equation:

$$P_r[\mathrm{SUCC}_0] = P_r[\mathrm{SUCC}_1]. \tag{4}$$

Game $G2$: By adding the SEND(S, M) oracle simulations, we converted game $G1$ to game $G2$. $G2$ models as an active attack. At this point, the attacker A is aiming to accept a modified message by deceiving a participant. After that, A chooses to find collisions by querying the hash oracle. However, all of the messages are associated with the identity and a random number. Therefore, while using the SEND(S, M) oracle, there is no collision. The birthday paradox provides us with the following equation:

$$\left|P_r[\mathrm{SUCC}_2] - P_r[\mathrm{SUCC}_1]\right| \leq \frac{q_h^2}{2|\mathrm{Hash}|}. \tag{5}$$

Game $G3$: Game $G2$ is converted to this game $G3$ by adding the simulations of the CorruptMD(u) oracles. Usually, the users tend to select the low entropy passwords and store the passwords on the mobile device. Thus, the attacker A tries to use the online dictionary attack to obtain the passcode. The system ought to restrict the quantity of incorrect password entries. So we have the following equation:

$$\left|P_r[\mathrm{SUCC}_3] - P_r[\mathrm{SUCC}_2]\right| \leq \frac{2q_{\mathrm{send}}}{|D|}. \tag{6}$$

Finally, each random oracle is simulated. The only way the attackers can succeed in the game after consulting the Test(u/S) oracle is to guess the bit. We have the following equation:

$$P_r[\mathrm{SUCC}_3] = \frac{1}{2}. \tag{7}$$

From the above games, we have $\mathrm{Adv}_{P1}^{\mathrm{Ake}}(A) = 2\Pr[\mathrm{SUCC}_0] - 1$, $P_r[\mathrm{SUCC}_0] = P_r[\mathrm{SUCC}_1]$, $|P_r[\mathrm{SUCC}_2] - P_r[\mathrm{SUCC}_1]| \le q_h^2/2|\mathrm{Hash}|$, $|P_r[\mathrm{SUCC}_3] - P_r[\mathrm{SUCC}_2]| \le (2q_s$ end$/|D|)$, and $P_r[\mathrm{SUCC}_3] = 1/2$. Thus, we can conclude that

$$\mathrm{Adv}_{P1}^{\mathrm{Ake}}(A) \le \frac{q_h^2}{|\mathrm{Hash}|} + \frac{2q_{\mathrm{send}}}{|D|}. \tag{8}$$

According to the analysis of $G0$ to $G3$, we can confirm that the suggested protocol provides semantic security in our security model. □

5. Comparison and Performance Analysis

5.1. Security Analysis

5.1.1. Mutual Authentication. Verifying the identity between users and servers is a fundamental protocol procedure. During the authentication stage, user U sends message $\{\mathrm{TID}_m, Q, P\}$ to server S. Then, S can authenticate user U by checking equation $P = H(\mathrm{PIM}_m \| r_0')$. If it holds, it means that user U is legal because only legal user U can compute PID_m and send the random number r_0 to server S. After that, S sends message $\{Z_1, Z_2\}$ to user U. When the message from server S is received, user U authenticates server S by checking equation $Z_1 = Z_2$. If the equation is valid, it means that U is legal. If it is not, it means that U is illegal because only the legal U can receive the secret random number r_s and compute the correct Z_1. Our protocol can therefore provide mutual authentication.

5.1.2. Perfect Forward Secrecy. The session key in our proposed protocol is $\mathrm{SK}' = H(r_0 r_s' \| \mathrm{TID}_m \| R_m')$, which is generated from the hash function $H(\cdot)$ and the secret random numbers r_0 and r_s. User U and server S chose different secret random numbers in every session. Therefore, even if an attacker obtains a subset of the session key and the master key of the server, he or she cannot guess any other session key due to the lack of the secret random number r_0 generated from user U or secret random number r_s generated from server S. Despite applying for hundreds of jobs, the attacker still cannot compute the session key SK. As a result, our protocol can resist perfect forward secrecy.

5.1.3. Resistance to Impersonation Attacks. If an adversary attempts to gain access to the remote server S for services, they will masquerade themselves as a legitimate mobile device. However, attackers cannot generate message $\{\mathrm{TID}_m, Q, P\}$ to pass the server's authentication. Server S can authenticate user U by checking equation $P = H(\mathrm{PID}_m \| r_0')$. Only the legal user U possesses the right secret random number r_0. Therefore, attacker A cannot masquerade as legal user U to access remote server S for services. Similarly, user U can authenticate the server S by checking equation $Z_2 = H(r_s \| r_0' \| R_m'')$. Due to the random number r_s being generated by the legal server S, the authentication failed. Hence, our protocol can provide resistance to impersonation attacks.

5.1.4. Resistance to Stolen Verifier Table Attack. The identifying information $\{\mathrm{TID}_m, \mathrm{PID}_m, T_m\}$ of user U is stored on server S. Since the user's identity is connected with the secret random number r_i, they are effectively hidden by the secure one-way hash function $H(\cdot)$. The attacker A derives the user's identity ID from the equation, which is computationally infeasible, without the secret random number r_i being unknown. As a result, our protocol can overcome stolen verifier table attacks.

5.1.5. Resistance to the Denial-of-Service Attacks. In general, most protocols are degraded by denial-of-service (DOS), which causes authentication between servers and clients to fail. Their server computes a large number of tanglesome operations, such as dot product, and group operations. Nonetheless, our protocol only uses some simplified operations, such as one-way hash functions, string concatenation operations, and *XOR* operations, and the server does not need to calculate many computation—consuming operations. Hence, our protocol can perform well in resisting denial-of-service attacks.

5.1.6. Provide User Anonymity. We chose to pseudo the user's identity ID by computing $\mathrm{TID}_m = H(\mathrm{PID}_{\oplus} w_s)$ and $\mathrm{TID}_m = H(H(\mathrm{ID}_m r_i) \oplus w_s)$ instead of transmitting ID to the server directly through the unsecured open channel. If an attacker tends to steal the user's identity ID_m, first, he or she should acquire PID_m. Only the legal user U and server S have the right random numbers r_i and w_s. The adversaries cannot guess the random numbers r_i and w_s in polynomial time, so he or she is unable to compute TID_m. Therefore, our protocol can guarantee user anonymity.

5.1.7. Man-in-the-Middle Attack. Man-in-the-middle attack is a type of active eavesdropping attack. However, from the above analysis, the attacker A obtains the right secret random number r_0, and R_s is impractical. Therefore, an attacker A cannot pass the server's or user's authentication and camouflage to be a legal user U or legal server S. Thus, man-in-the-middle attacks can be resisted by our protocol.

5.1.8. Password Guessing Attack Resistance. In general, an attacker tries to guess the user's password by intercepting messages through public channels or by stealing the user's device. Our protocol transmits TID_m, Q, P and $\{Z_1, Z_2\}$ via insecure open channels. Nevertheless, $\{Z_1, Z_2\}$ contains nothing about the user's password. From analyzing $\{\mathrm{TID}_m, Q, P\}$, we have $Q = h_m \oplus H(T_m' \oplus w_s) \oplus T_m'$ and $T_m = H(\mathrm{PW}_m \| R_i \oplus H(\mathrm{ID}_m \oplus \mathrm{PW}_m))$. However, attackers

Table 2: Security comparisons.

	Our scheme	Debiao et al. [17]	Odelu et al. [26]	Farash and Attari [19]	Mo et al. [31]
Mutual authentication	Y	Y	Y	Y	Y
Resist inside attack	Y	N	Y	Y	Y
Without complex calculation	Y	N	Y	N	N
User anonymity	Y	N	Y	Y	Y
Resist impersonation attack	Y	N	Y	Y	Y
Resistance the denial-of-service attacks	Y	Y	N	N	N
Provide user anonymity	Y	N	Y	N	N
Resistance to password guessing attack	Y	Y	Y	Y	N

Table 3: Performance analysis notations.

Notations	Descriptions
T_{MAC}	Running time of message authentication code
T_H	Running time of the one-way-hash function
T_{IN}	Running time of modular inversion
T_{PM}	Running time of scalar multiplication operation of point

Table 4: Computational costs.

	Odelu et al. [26]	Farash and Attari [19]	Debiao et al. [17]	Mo et al. [31]	Our protocol
Computational cost (client)	$13T_H$	$3T_{\text{PM}} + 4T_H$	$3T_{\text{PM}} + 2T_H + 2\text{TMAC}$	$3T_{\text{PM}} + 2T_H$	$10T_H$
Running time (client) (ms)	0.13	205.16	183.70	372.11	0.12
Computational cost (server)	$24T_H$	$3T_{\text{PM}} + 6T_H$	$3T_{\text{PM}} + T_{\text{IN}} + 3T_H + 2T\text{Mac}$	$3T_{\text{PM}} + 2T_{\text{IN}} + 4T_H$	$7T_H$
Running time (server) (ms)	297.00	579.21	552.47	898.00	124.05

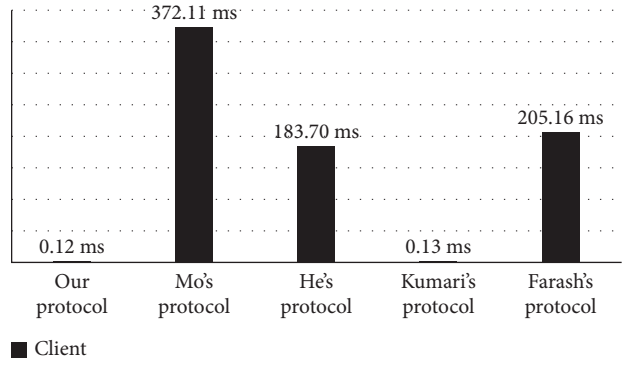

Figure 4: Computation expense of client.

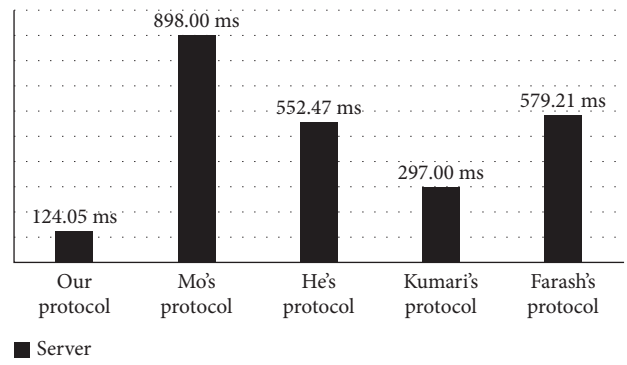

Figure 5: Computation expense of server.

cannot have both PW_m and ID_m at the same time. Similarly, we can prove that the attacker also cannot guess the password by stealing the user's device.

5.2. Performance Analysis. According to the above analysis (Section 4 security proof and Section 5.1 security analysis), our protocol can resist inside attacks, and impersonation attacks and can provide users with anonymity and mutual authentication. In our proposed scheme, we do not use any kind of complex operations, only some simple operations. In Table 2, we can see that the proposed protocol and other protocols are satisfied with mutual authentication. Both Debiao et al.'s [17] and Mo et al.'s [31] schemes contain complex calculations, while their protocols cannot provide user anonymity. The denial-of-service attacks has been found in Odelu et al.'s [26] and Mo et al.'s protocol [31]. In addition, in Mo et al.'s [31], Debiao et al.'s [17], and Farash and Attari's [19] protocol, the message's client ID was displayed in plaintext. Consequently, their scheme could not protect the anonymity of users' identities, especially in wireless network forms.

In this section, we analyzed the performance between our protocol and the schemes proposed by other scholars. We compared all the protocols in a practical environment with the ones that appear in this article, which are built using a standard cryptographic library named MIRACL [40]. Our

computer runs a Windows 10 Pro (64 bits) operating system with an Intel (R) Core (TM) i5-7300HQ @ 2.50-GHz processor and a 8-GB memory. Table 3 describes the notation used in this phrase.

In Table 4, we showed how our suggested protocol compares to those of other researchers. String concatenation operations and XOR operations are considerably less computationally expensive than elliptic curve point multiplication operations, modular multiplication operations, bilinear pairing operations, and one-way hash functions. As a result, we ignored the XOR and string concatenation operations in Table 4.

Figures 4 and 5 summarized the running time comparison results. In Figure 4, we can see that the proposed scheme has less computing time than the other protocols. Therefore, our protocol has a fast response speed on the client side. In Figure 5, we can see that our proposed protocol has very little demand for computing resources from clients. Although the response times of Odelu et al.'s [26] protocol are similar to our proposed scheme on the client side, their protocol has more computing time. The proposed protocol can perform better than any other protocol in a practical environment. Our suggested protocol is therefore better suited for mobile terminals.

6. Conclusion

To safeguard users' privacy, we designed a novel high-efficiency mutual-authentication and key agreement protocol for the mobile client-server environment. Our protocol only employs a few basic operations, including XOR, one-way hashing, and string concatenation. The proposed protocol, which can operate on various types of mobile terminals, is capable of achieving the same security level with high efficiency and using fewer computing resources than the related work. The random oracle model also demonstrates the security of the suggested protocol. According to the analysis above, we can infer that our protocol can satisfy the requirements for response time and security in mobile client-server environments.

Data Availability

The data used to support the findings of this study are available from the first author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key Foundation for Exploring Scientific Instrument of China (2013YQ03065104).

References

[1] A. Fabio, "Smishing and the rise of mobile banking attacks," 2021, https://securelist.com/smishing-and-the-rise-of-mobile-banking-attacks/75575/.

[2] V. S. Miller, "Use of elliptic curves in cryptography," in *Proceedings of the Lecture notes in computer sciences 218 on Advances in cryptology—CRYPTO 85*, Santa Barbara, CA, USA, June 1986.

[3] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.

[4] M. Shand and J. Vuillemin, "Fast implementations of RSA cryptography," in *Proceedings of the IEEE 11th Symposium on Computer Arithmetic*, pp. 252–259, Windsor, ON, Canada, June 1993.

[5] Y. Tsiounis and M. Yung, "On the security of ElGamal based encryption," in *Proceedings of the International Workshop on Public Key Cryptography*, pp. 117–134, Pacifico Yokohama, Japan, February 1998.

[6] S. Halevi and H. Krawczyk, "Public-key cryptography and password protocols," *ACM Transactions on Information and System Security*, vol. 2, no. 3, pp. 230–268, 1999.

[7] Y. Wang, "Public key cryptography standards: PKCS," in *Proceedings of the Advances in Cryptology – EUROCRYPT '90*, Aarhus, Denmark, May 2012.

[8] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *SIAM Journal on Computing*, vol. 32, no. 3, pp. 586–615, 2003.

[9] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[10] L. H. Li, L. C. Lin, and M. S. Hwang, "A remote password authentication scheme for multiserver architecture using neural networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1498–1504, 2001.

[11] W. S. Juang, "Efficient multi-server password authenticated key agreement using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 251–255, 2004.

[12] C. C. Chang and J. S. Lee, "An efficient and secure multi-server password authentication scheme using smart cards," in *Proceedings of the 2004 international conference on cyberworlds*, pp. 417–422, Tokyo, Japan, November 2004.

[13] W. J. Tsaur, J. H. Li, and W. B. Lee, "An efficient and secure multi-server authentication scheme with key agreement," *Journal of Systems and Software*, vol. 85, no. 4, pp. 876–882, 2012.

[14] Y. P. Liao and S. S. Wang, "A secure dynamic ID based remote user authentication scheme for multi-server environment," *Computer Standards & Interfaces*, vol. 31, no. 1, pp. 24–29, 2009.

[15] H. C. Hsiang and W. K. Shih, "Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment," *Computer Standards & Interfaces*, vol. 31, no. 6, pp. 1118–1123, 2009.

[16] C. C. Lee, T. H. Lin, and R. X. Chang, "A secure dynamic ID based remote user authentication scheme for multi-server environment using smart cards[J]," *Expert Systems with Applications*, vol. 38, no. 11, pp. 13863–13870, 2011.

[17] H. Debiao, C. Jianhua, and H. Jin, "An ID-based client authentication with key agreement protocol for mobile client–server environment on ECC with provable security," *Information Fusion*, vol. 13, no. 3, pp. 223–230, 2012.

[18] D. Wang and C. Ma, "Cryptanalysis of a remote user authentication scheme for mobile client–server environment based on ECC," *Information Fusion*, vol. 14, no. 4, pp. 498–503, 2013.

[19] M. S. Farash and M. A. Attari, "A secure and efficient identity-based authenticated key exchange protocol for mobile

client–server networks," *The Journal of Supercomputing*, vol. 69, no. 1, pp. 395–411, 2014.

[20] S. Jegadeesan, M. Azees, P. M. Kumar et al., "An efficient anonymous mutual authentication technique for providing secure communication in mobile cloud computing for smart city applications," *Sustainable Cities and Society*, vol. 49, Article ID 101522, 2019.

[21] A. Irshad, S. A. Chaudhry, M. Shafiq, M. Usman, M. Asif, and A. Ghani, "A provable and secure mobile user authentication scheme for mobile cloud computing services," *International Journal of Communication Systems*, vol. 32, no. 14, Article ID e3980, 2019.

[22] O. Olufemi Olakanmi and S. O. Oke, "MASHED: security and privacy-aware mutual authentication scheme for heterogeneous and distributed mobile cloud computing services," *Information Security Journal: A Global Perspective*, vol. 27, no. 5-6, pp. 276–291, 2018.

[23] J. L. Tsai and N. W. Lo, "A privacy-aware authentication scheme for distributed mobile cloud computing services," *IEEE Systems Journal*, vol. 9, no. 3, pp. 805–815, 2015.

[24] A. Irshad, M. Sher, and H. F. Ahmad, "An improved multi-server authentication scheme for distributed mobile cloud computing services[J]," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 10, no. 12, pp. 5529–5552, 2016.

[25] H. Jannati and B. Bahrak, "An improved authentication protocol for distributed mobile cloud computing services," *International Journal of Critical Infrastructure Protection*, vol. 19, pp. 59–67, 2017.

[26] V. Odelu, A. K. Das, S. Kumari, X. Huang, and M. Wazid, "Provably secure authenticated key agreement scheme for distributed mobile cloud computing services," *Future Generation Computer Systems*, vol. 68, pp. 74–88, 2017.

[27] I. S. K. Hafizul and G. P. Biswas, "An improved ID-based client authentication with key agreement scheme on ECC for mobile client-server environments," *Theoretical and Applied Informatics*, vol. 24, 2012.

[28] H. Sun, Q. Wen, H. Zhang, and Z. Jin, "A novel remote user authentication and key agreement scheme for mobile client-server environment," *Applied Mathematics & Information Sciences*, vol. 7, no. 4, pp. 1365–1374, 2013.

[29] D. He and D. Wang, "Robust biometrics-based authentication scheme for multiserver environment," *IEEE Systems Journal*, vol. 9, no. 3, pp. 816–823, 2015.

[30] V. Odelu, A. K. Das, and A. Goswami, "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1953–1966, 2015.

[31] J. Mo, Z. Hu, and Y. Lin, "Remote user authentication and key agreement for mobile client–server environments on elliptic curve cryptography," *The Journal of Supercomputing*, vol. 74, no. 11, pp. 5927–5943, 2018.

[32] Y. M. Tseng, S. S. Huang, T. T. Tsai, and J. H. Ke, "List-free ID-based mutual authentication and key agreement protocol for multiserver architectures," *IEEE Transactions on Emerging Topics in Computing*, vol. 4, no. 1, pp. 102–112, 2016.

[33] M. Azrour, J. Mabrouki, A. Guezzaz, and Y. Farhaoui, "New enhanced authentication protocol for Internet of Things," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 1–9, 2021.

[34] M. Wazid, A. K. Das, V. Bhat K, and A. V. Vasilakos, "LAM-CIoT: lightweight authentication mechanism in cloud-based IoT environment," *Journal of Network and Computer Applications*, vol. 150, Article ID 102496, 2020.

[35] S. A. Chaudhry, A. Irshad, K. Yahya, N. Kumar, M. Alazab, and Y. B. Zikria, "Rotating behind privacy: an improved lightweight authentication scheme for cloud-based IoT environment," *ACM Transactions on Internet Technology*, vol. 21, no. 3, pp. 1–19, 2021.

[36] X. Wang, W. Guo, and W. Zhang, "Cryptanalysis and improvement on a parallel keyed hash function based on chaotic neural network," *Telecommunication Systems*, vol. 52, no. 2, pp. 515–524, 2013.

[37] M. Abdalla, P. A. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," *International Workshop on Public Key Cryptography*, Springer, Berlin, Heidelberg, 2005.

[38] C. C. Chang and H. D. Le, "A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 357–366, 2016.

[39] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Transactions on Computers*, vol. 51, no. 5, pp. 541–552, 2002.

[40] Shamus Software Ltd, "Miracl library," 2022, http://www.shamus.ie/index.php?page=home.